# Investigating the effects of retail sales promotions with association rule learning


by Giacinto Romano, BSc Computer Science
supervised by Dr. Ilias Kotsireas
Wilfrid Laurier University


23rd March 2018


## Abstract

This project is an extension of Market Basket Analysis to apply to items on promotion. When a grocery retailers place an item on promotion the lower sales price in that period is expected to lead to higher volume of sales for that product.

However, there is also a cannibalization (items being negatively affected) and halo effect (items being positively affected) that is assumed to occur with items related to the promoted item. For example, if hot dogs go on sale not only will it sell in higher volume but sausages would be expected to sell less (cannibalization) and hot dog buns would be expected to sell higher volume.



[1]

The goal of this project is to use association rule learning algorithms to create definitive cause-and-effect rules for relationships between items. Specifically, the Apriori Algorithm will be used to determine the scale of the Halo and Cannibilization effect relating to Items on promotion. The final product will include a program able to translate sales data into an item-set and that item-set into a set of association rules

## 1 Creating the Item-Set

The first step of this project requires the creation of an item-set that is used by the Apriori Algorithm to create rules. This item-set is a group of transactions and their corresponding items for a given time frame and in this

### 1.1 Encoding the Items

Each week at a given location is defined as a transaction and the items for each transaction are encoded with a bit relating to sales performance and whether or not an item was on promotion.

| Encoded Bit | Meaning |
|---|---|
| .P | Item on promotion that week |
| .X | Item experienced expected level of sales that week |
| .O | Item experienced higher than expected sales that week |
| .U | Item experienced lower than expected sales that week |

If an item is on promotion it will have a P as its bit regardles of its sales performance that week.

Whether an item has expected (X), under (U) or oversold (O) will be based on a given weeks variance (V) from historic sales. The variance threshold will be an input from the user such that in order to be deemed expected sales the followeing must hold:

$$\left|\frac{(x-\bar{x})}{\bar{x}}\right| \leq V$$

$$\bar{x}: sales\ average$$

$$x: period\ sales$$

$$V: Variance\ (0 < V < 1)$$

Else, the item will be encoded with with either the under (U) or oversold (O) bit.

### 1.2 Finalizing the Item-Set

The logic described in the previous sections will be applied in the following manner:

| Item | Week | Promo_Flag | $x$ | $\bar{x}$ |
|---|---|---|---|---|
| Eggs | 5 | 0 | 350 | 250 |
| Bacon | 5 | 1 | 500 | 300 |
| Sausage | 5 | 0 | 300 | 400 |
| Toothpaste | 5 | 0 | 275 | 250 |
| Honey | 6 | 0 | 150 | 200 |
| Jam | 6 | 1 | 300 | 200 |
| PeanutButter | 6 | 0 | 250 | 200 |
| Soap | 6 | 0 | 250 | 250 |

The above table is converted to the following item-set, V = 0.1 as set by program user:

| Transaction_ID | Items |
|---|---|
| S1W5 | Eggs.O, Sausage.U, Bacon.P, Toothpaste.X |
| S1W6 | Honey.U, Jam.P, PeanutButter.O, Soap.X |

## 2 Apriori Algorithm

Apriori works by creating frequent item-sets with that have a minimum level of support and using the item-sets to create strong association rules that have a minimum level of confidence. [4]

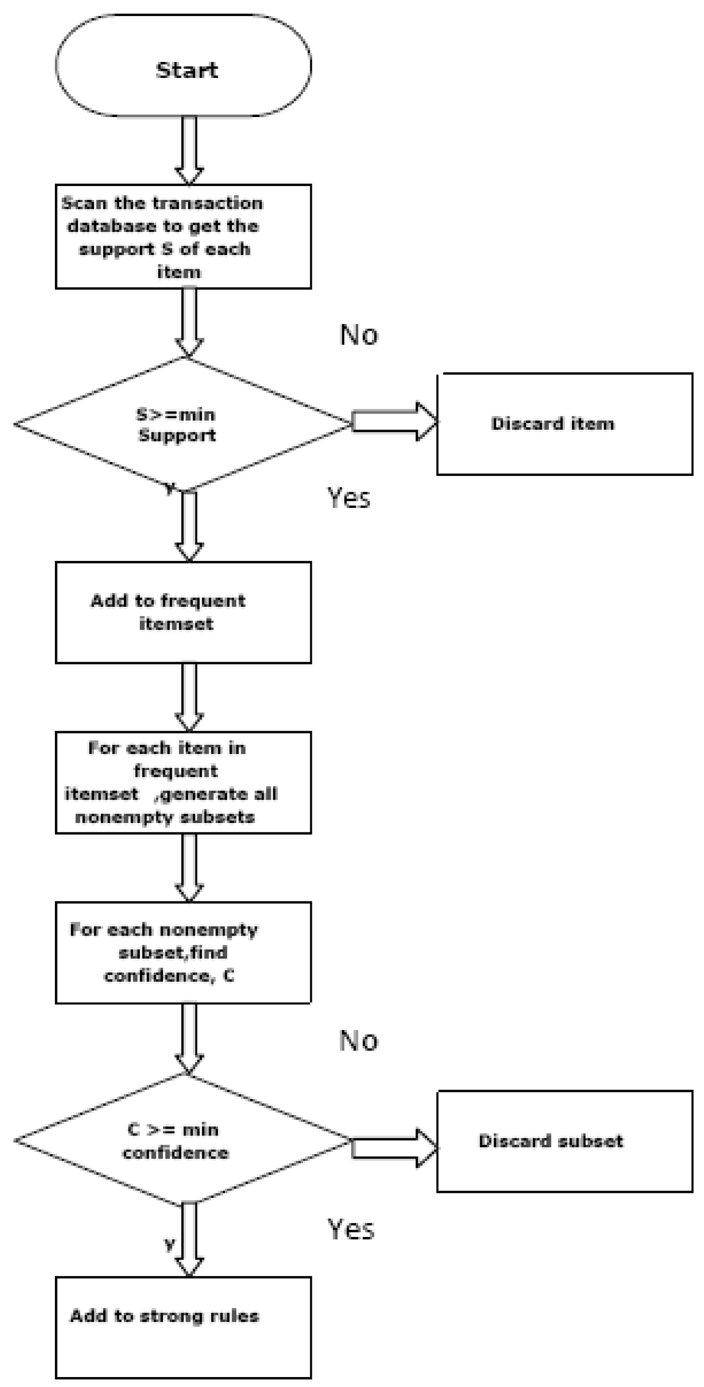
### 2.1 Support and Confidence

Minimum Support Needed (S) : The minimum numbers of times an item-set must occur to be significant enough to be considered for creating rules

Minimum Confidence Needed (C): The minimum number of times a rule must be found for it to be significant. For example, if eggs→bacon is the rule then there must be a minimum proportion, C, of the transactions that contain eggs which also contain bacon.

### 2.2 The Process of Apriori

Afterthe user of the program determines the minimum level of support and confidence necessary, the algorithm will progress using the steps outlined below. [2]




### 2.3 Pseudocode

```
Ck: Candidate itemset of size k
Lk : frequent itemset of size k

L1 = {frequent items};
for (k = 1; Lk !=∅; k++) do begin
    Ck+1 = candidates generated from Lk;
   for each transaction t in database do
        increment the count of all candidates in Ck+1
      that are contained in t
   Lk+1  = candidates in Ck+1 with min_support
   end
return ∪k Lk;
```

After the frequent item-sets of different length are created they are said to be valid rules if they the item-set has a valid level of confidence [3].

## Conclusion

The project can be summarized in the following manner:

1. Preprocessing.

   Input: Item sales data, variance amount
   Output: Item-set

2. Apriori Algorithm

   Inputs: Item-set, Confidence Level, Support Level
   Output: Strong Rules from frequent itemsets

## References


[1] - Market Basket Analysis. Data Mining (2014). Available at: https://uwsdatamining.wordpress.com/market-basket-analysis/.

[2] - Mitta, M., Singh, J., Aggarwal, A., Kumari, K. & Yadav, M. Ordering Policy for Imperfect Quality Itemsets using Cross Selling Effects. (2014).

[3] - Wasilewska, A. Apriori Algorithm. Available at: http://www3.cs.stonybrook.edu/~cse634/lecture_notes/07apriori.pdf.

[4] - Tan, P.-N., Steinbach, M. & Kumar, V. Introduction to data mining. (Pearson, 2015).